Request Tracking

Carl Solomon

csolomon@eos.hitc.com

1 November 1995

Request Tracking Overview

Overview/Design Drivers:

- Track the status of requests through the system from end-to-end in near real-time
- Tracked Requests (EcRequest) are limited to:
  - User-requests (e.g., product, subsetting)
  - Ingest requests
  - Selected DAAC Operator requests (archive backup request)
- Track requests across multiple DAACs
- Allow User Services to get status of user-requests on behalf of user and allow User to get status on their own requests
- Provide standard reporting of requests for User Services and M&O use

ECS Context

- System wide

Scenario Context

- Pull scenarios (Quick Access, Coincident Search, Disconnected Session)

Request Tracking Design

EcRequest is a distributed object provided by MSS.

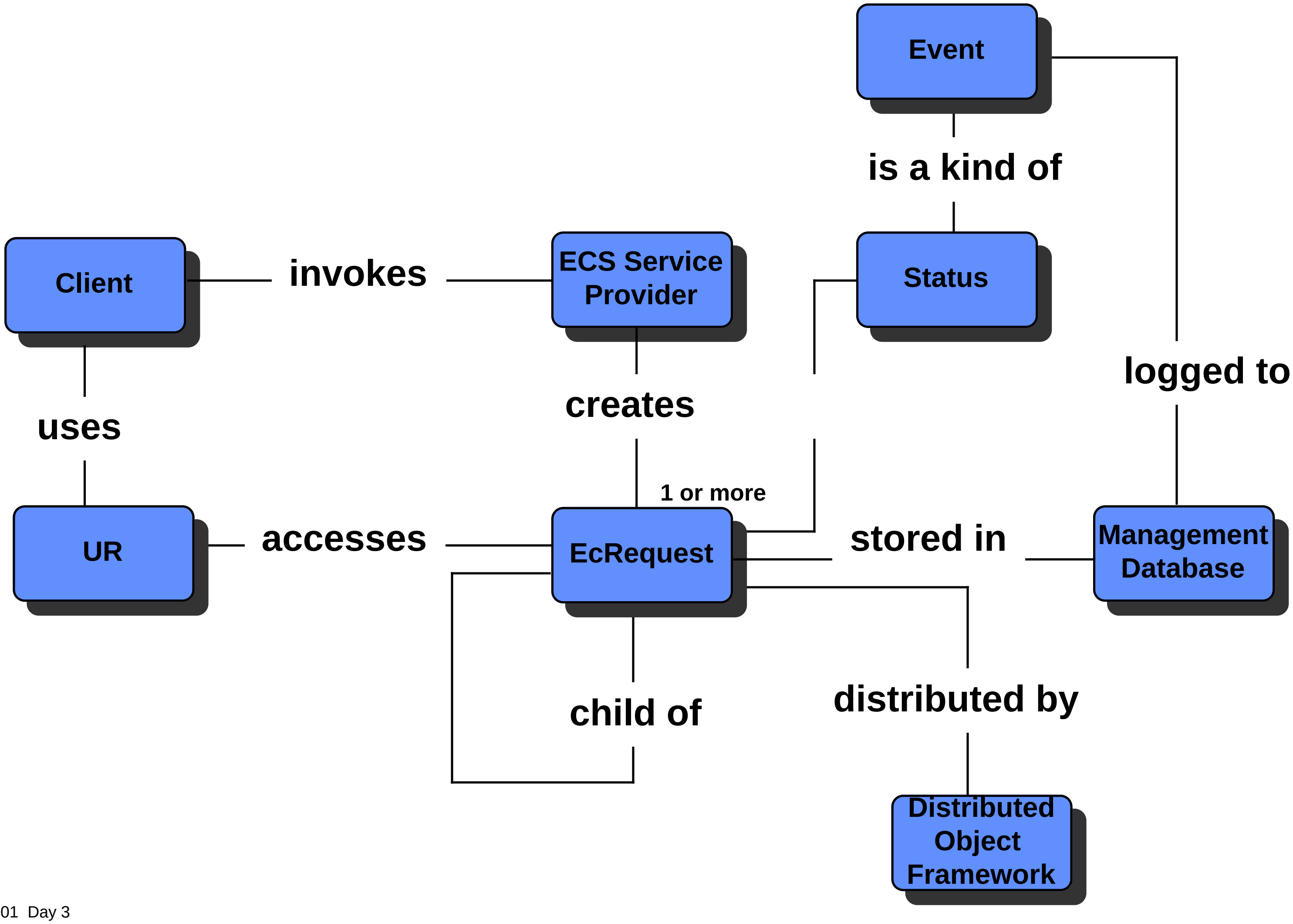
EcRequest is represented as a Universal Reference (UR) and is unique across the system.

EcRequest is updated by those subsystems acting on that Request with the latest status.

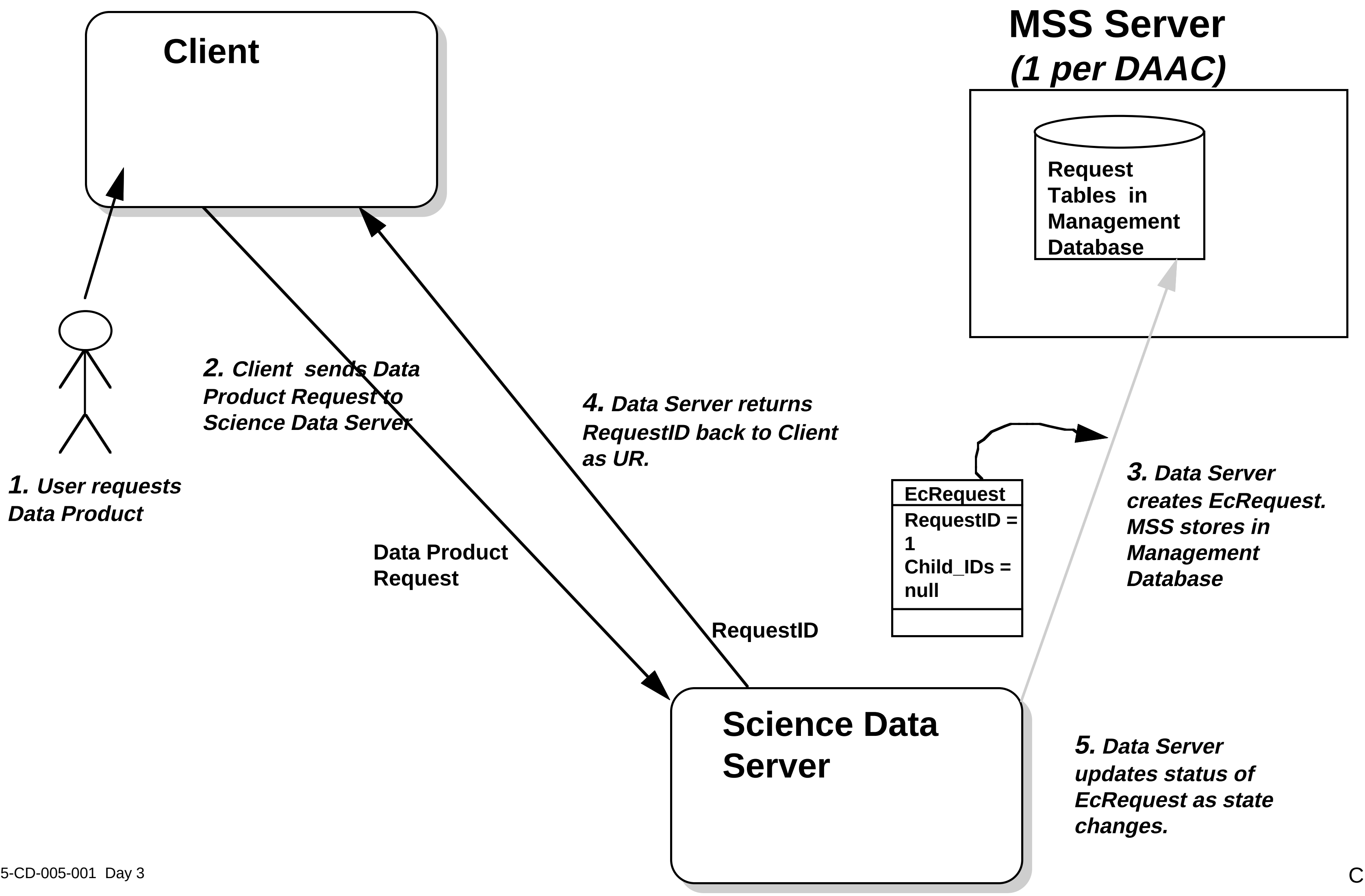
Lifetime of EcRequest is beyond actions on it by a single-subsystem. (persists in MSS Management Database)

EcRequests can be tracked across DAAC boundaries.

Software Design -
High Level Class Model



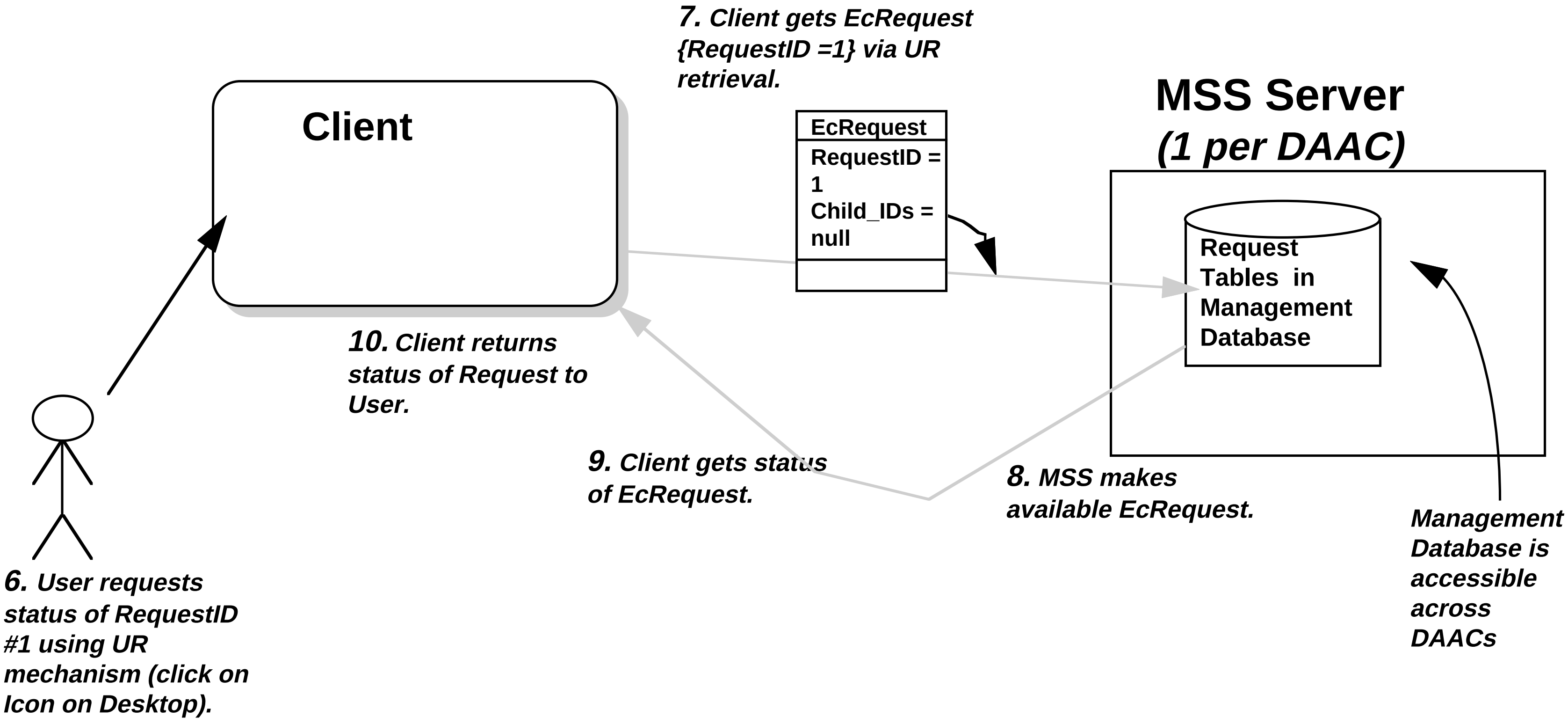
Single Data Server Example – Request Issued



Client
MSS Server
(1 per DAAC)
Request Tables in Management Database
2. Client sends Data Product Request to Science Data Server
4. Data Server returns RequestID back to Client as UR.
1. User requests Data Product
EcRequest
RequestID = 1
Child_IDs = null
3. Data Server creates EcRequest. MSS stores in Management Database
Data Product Request
RequestID
Science Data Server
5. Data Server updates status of EcRequest as state changes.

Single Data Server Example
– User Desires Status of Request

Assumption: Session lifetime has timed out or client disconnected



7. Client gets EcRequest {RequestID =1} via UR retrieval.

10. Client returns status of Request to User.

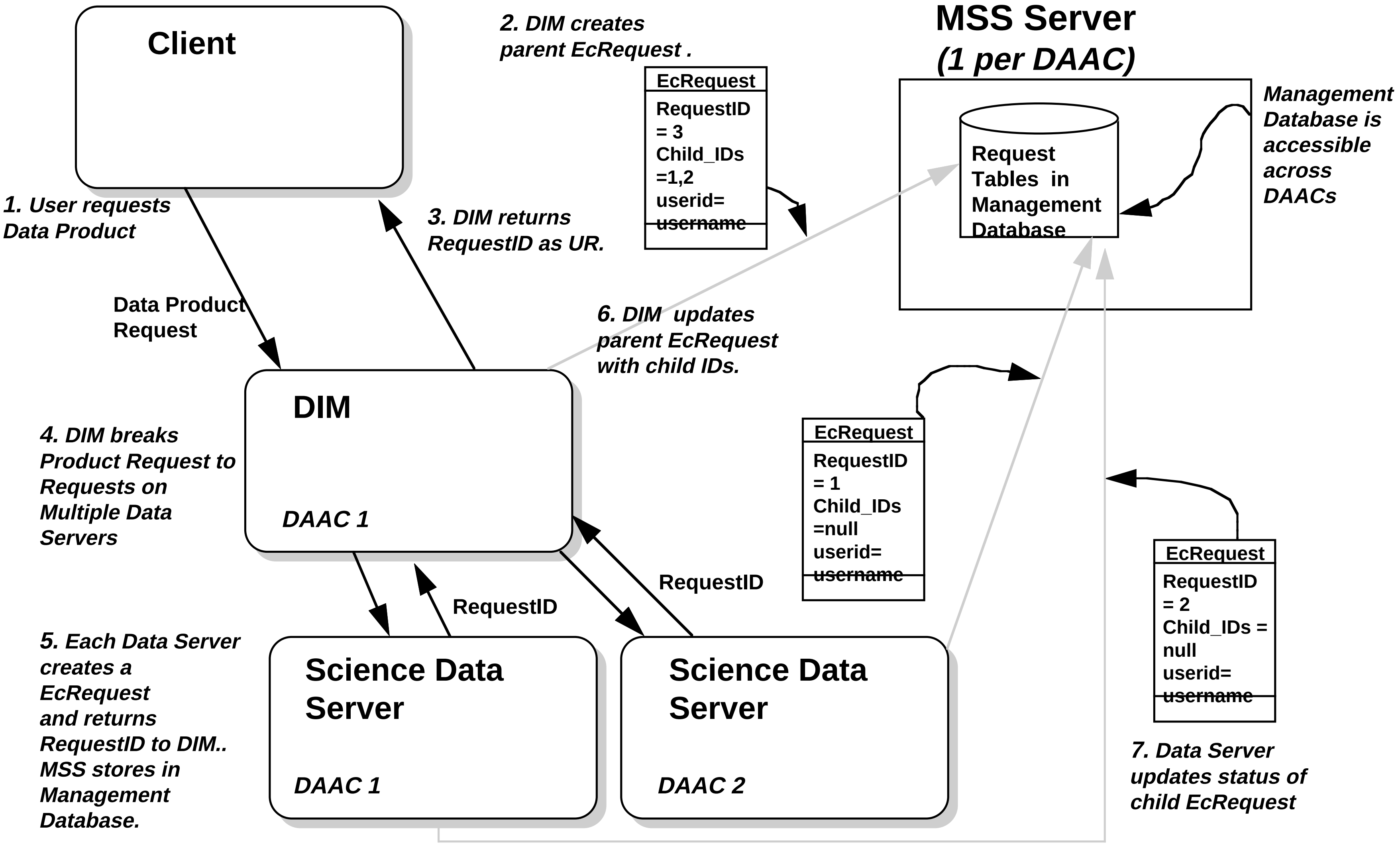
9. Client gets status of EcRequest.

8. MSS makes available EcRequest.

6. User requests status of RequestID #1 using UR mechanism (click on Icon on Desktop).

Management Database is accessible across DAACs

Multiple Data Server Example – Request Issued



Client

1. User requests Data Product

Data Product Request

2. DIM creates parent EcRequest .

EcRequest

RequestID = 3
Child_IDs =1,2
userid= username

3. DIM returns RequestID as UR.

MSS Server
(1 per DAAC)

Request Tables in Management Database

Management Database is accessible across DAACs

DIM

DAAC 1

4. DIM breaks Product Request to Requests on Multiple Data Servers

6. DIM updates parent EcRequest with child IDs.

EcRequest

RequestID = 1
Child_IDs =null
userid= username

RequestID

RequestID

EcRequest

RequestID = 2
Child_IDs = null
userid= username

5. Each Data Server creates a EcRequest and returns RequestID to DIM.. MSS stores in Management Database.

Science Data Server

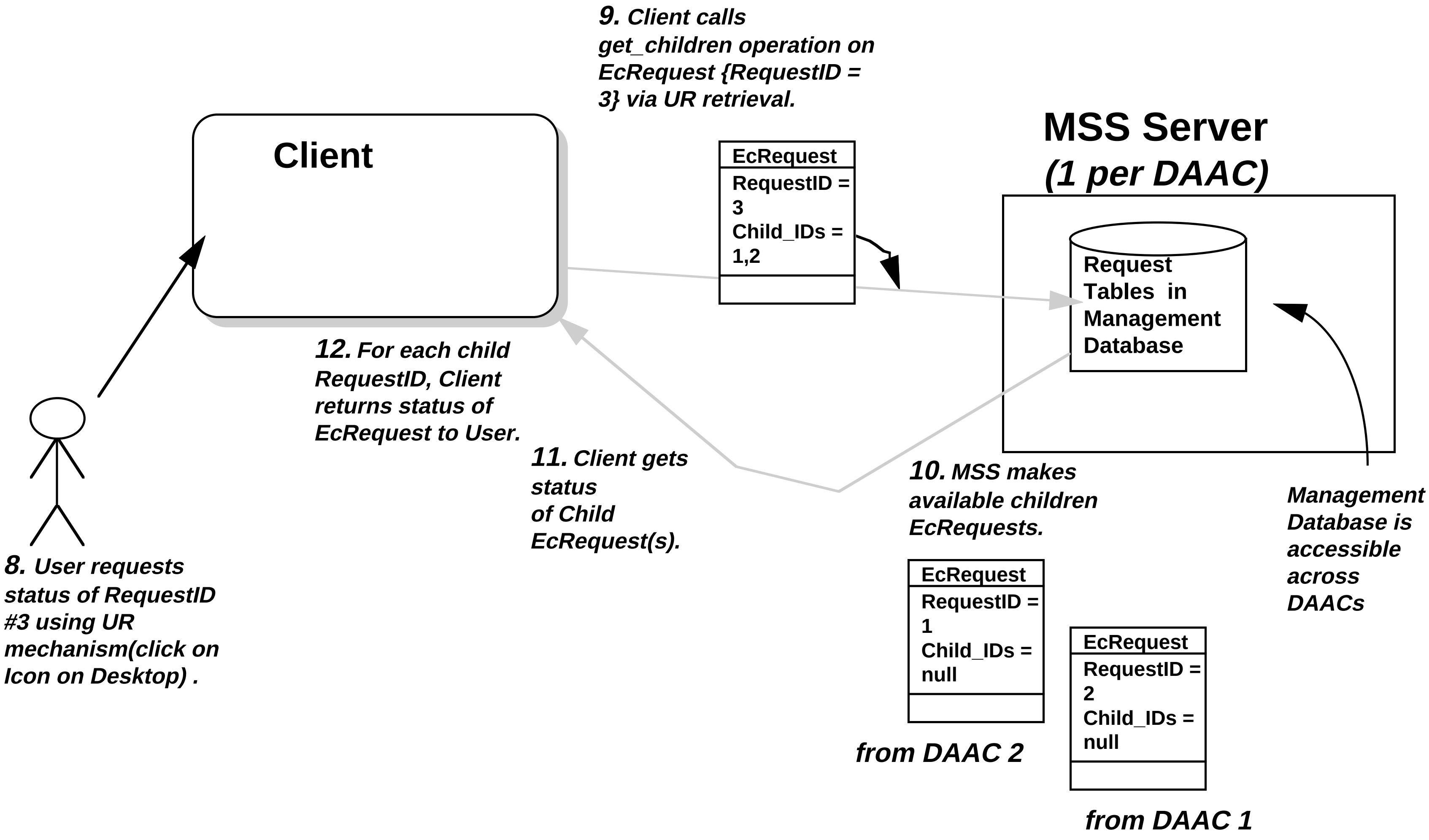
DAAC 1

Science Data Server

DAAC 2

7. Data Server updates status of child EcRequest

Multiple Data Server Example
– User Desires Status of Request



Client

MSS Server

(1 per DAAC)

8. User requests status of RequestID #3 using UR mechanism(click on Icon on Desktop) .

9. Client calls get_children operation on EcRequest {RequestID = 3} via UR retrieval.

EcRequest

RequestID = 3

Child_IDs = 1,2

Request Tables in Management Database

10. MSS makes available children EcRequests.

EcRequest

RequestID = 1

Child_IDs = null

from DAAC 2

EcRequest

RequestID = 2

Child_IDs = null

from DAAC 1

Management Database is accessible across DAACs

11. Client gets status of Child EcRequest(s).

12. For each child RequestID, Client returns status of EcRequest to User.

Current Status/Plan for CDR

Potential Future Enhancements

- Exploitation of emerging COTS capabilities

Next Steps

- Update MSS Server and Network hardware sizing to include EcRequests.

- Determine best mechanism for sharing EcRequest data across DAACs.